

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та роботехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)
Розроблення автоматичної системи трекінгу для комп'ютерного зору
квадрокоптера
(тема)

Виконав:
здобувач 2 року навчання, групи КТРСм-24-2
Владислав МАМІН
(власне ім'я, прізвище)

Спеціальність 174 Автоматизація,
комп'ютерно-інтегровані технології та
робототехніка
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Комп'ютеризовані та
роботехнічні системи
(повна назва освітньої програми)

Керівник проф. каф. Олександр ЦИМБАЛ
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри

(підпис)

Ігор НЕВЛЮДОВ
(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та роботехніки

Рівень вищої освіти другий(магістерський)

Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка

(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Комп'ютеризовані та роботехнічні системи
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____» _____ 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувачеві Маміну Владиславу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення автоматичної системи трекінгу для комп'ютерного зору квадрокоптера

затверджена наказом університету від 10 листопада 2025 р. № 1080Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 26 грудня 2025 р.

3. Вихідні дані до роботи Конструкції квадрокоптерів, системи трекінгу, технології комп'ютерного зору

4. Перелік питань, що потрібно опрацювати в роботі 4.1 Аналіз предметної області та постановка задачі 4.2 Квадрокоптери. Типи та використання 4.3 Огляд технічних засобів комп'ютерного зору 4.4 Висновки аналізу предметної області 4.5 Аналіз сучасних методів комп'ютерного зору 4.6 Системи комп'ютерного зору 4.7 Аналіз методів обробки зображень 4.8 Огляд інтелектуальних систем комп'ютерного зору 4.9 Висновки аналізу методів комп'ютерного зору 4.10 Програмна реалізація трекінгу об'єктів 4.11 Вибір програмної середи 4.12 Дослідження ефективності розробленої програми 4.13 Охорона праці 4.14 Висновок експериментальної частини 4.15 Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) презентація PowerPoint на 12 стор.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

| Найменування розділу | Консультант (посада, прізвище, ім'я, по батькові) | Позначка консультанта про виконання розділу | |
|----------------------|--|---|------|
| | | підпис | дата |
| | | | |
| | | | |

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи | Строк / терміни виконання етапів роботи | Примітка |
|---|--|---|----------|
| 1 | Аналіз технічного завдання | 28.11.2025 | виконано |
| 2 | Аналіз існуючих рішень та концепцій | 05.12.2025 | виконано |
| 3 | Вибір методів обробки зображень та алгоритмів трекінгу | 15.12.2025 | виконано |
| 4 | Розробка програмного рішення трекінгу об'єктів | 18.12.2025 | виконано |
| 5 | Проведення експериментальної частини | 23.12.2025 | виконано |
| 6 | Оформлення пояснювальної записки | 25.12.2025 | виконано |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Дата видачі завдання 10 листопада 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. каф. Олександр ЦИМБАЛ
(підпис) (посада, власне ім'я, прізвище)

Я, Мамін Владислав Андрійович, як здобувач вищої освіти ХНУРЕ, розумію і підтримую політику закладу з академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Я не використовував штучний інтелект для підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

“25” грудня 2025 року



Мамін В.А

РЕФЕРАТ

Пояснювальна записка: 79 стор., 45 рис., 3 дод., 15 джерел.

КВАДРОКОПТЕР, КОМП'ЮТЕРНИЙ ЗІР, ТРЕКІНГ ОБ'ЄКТІВ,
OPENCV, МЕТОДИ ОБРОБКИ ЗОБРАЖЕННЯ.

Мета кваліфікаційної роботи – розроблення трекінгу об'єктів з використанням комп'ютерного зору квадрокоптера шляхом розробки програмного забезпечення на базі C++ OpenCV.

Об'єкт дослідження – процес трекінгу об'єктів на основі комп'ютерного зору квадрокоптера.

Предмет дослідження – методи обробки зображень, ітераційний алгоритм CAMShift.

Методи дослідження – аналіз алгоритму обробки вхідного зображення CAMShift через комп'ютерний зір квадрокоптера для реалізації трекінгу об'єктів.

Результатом кваліфікаційної роботи є розробка програмної частини трекінгу об'єктів використовуючи методи обробки зображення бібліотеки OpenCV та алгоритму CAMShift. Розроблена програма має можливість відслідковувати над вибраною областю об'єкта у різних умовах освітленості.

Отримані результати можна віднести до Цілі сталого розвитку 11 “Сталий розвиток міст і громад”, а саме п.11.4 “Забезпечити своєчасне оповіщення населення про надзвичайні ситуації з використанням інноваційних технологій”.

ABSTRACT

Explanatory note: 79 pages, 45 fig., 3 app., 15 sources.

QUADCOPTER, COMPUTER VISION, OBJECT TRACKING, OPENCV, IMAGE PROCESSING METHODS.

The purpose of the qualification work is a development of object tracking using quadcopter computer vision by developing software based on C++ OpenCV.

The object of the study is the process of tracking objects based on the computer vision of a quadcopter.

The subject of the study is image processing methods, the iterative algorithm CAMShift.

The method of research is an analysis of the CAMShift image processing algorithm through the computer vision of a quadcopter to implement object tracking.

The result of master's thesis is the development of a software part for tracking objects using the image processing methods of the OpenCV library and CAMShift algorithm. The developed program has the ability to track over a selected area of object in different lighting conditions.

The results obtained can be attributed to Sustainable Development Goal 11 "Sustainable development of cities and communities", namely item 11.4 "Ensure timely notification of the population about emergencies using innovative technologies".

ЗМІСТ

| | |
|---|----|
| Перелік скорочень | 8 |
| Вступ..... | 9 |
| 1 Аналіз предметної області та постановка задачі | 11 |
| 1.1 Постановка задачі кваліфікаційної роботи..... | 11 |
| 1.2 Огляд технічних засобів комп'ютерного зору | 18 |
| 1.3 Висновки аналізу предметної області | 21 |
| 2 Аналіз сучасних методів комп'ютерного зору..... | 22 |
| 2.1 Системи комп'ютерного зору | 22 |
| 2.2 Аналіз методів обробки зображень | 24 |
| 2.3 Огляд інтелектуальних систем комп'ютерного зору..... | 31 |
| 2.4 Застосування методів систем комп'ютерного зору | 32 |
| 2.5 Висновки аналізу методів комп'ютерного зору..... | 33 |
| 3 програмна реалізація трекінгу об'єктів..... | 34 |
| 3.1 Вибір програмної середи | 34 |
| 3.2 Програмна реалізація трекінгу об'єкту | 34 |
| 3.3 Дослідження ефективності розробленої програми | 41 |
| 3.4 Охорона праці | 49 |
| 3.5 Висновок експериментальної частини | 50 |
| Висновки | 51 |
| Перелік джерел посилання | 52 |
| Додаток А Код програми..... | 54 |
| Додаток Б Висвітлення результатів у статті..... | 60 |
| Додаток В Демонстраційний матеріал у вигляді презентації..... | 66 |

ПЕРЕЛІК СКОРОЧЕНЬ

АКБ – акумуляторна батарея;

БПЛА – безпілотний літальний апарат;

ЗП – зарядний пристрій;

САК – система автоматичного керування.

FPV – First Person View;

GPS – Global Positioning System;

IMU – Inertial Measurement Unit;

LIDAR – Light Detection and Ranging.

ВСТУП

З розвитком цифрових технологій в світі збільшується актуальність в використанні алгоритмів обробки зображень, нейронних мереж та штучного інтелекту.

Комп'ютерний зір – це поняття обробки, аналізу зображень які використовуються для машинного зору робота або пристрою. Застосовується в більшості сучасних розробках: системи взаємодії, відеоспостереження або систем керування процесами.

На сьогоднішній день з'являється безліч різних видів квадрокоптерів які використовують комп'ютерний зір в виконанні задач по контролю місцевості, постає задача в спрощенні у використанні та автоматизації більшості задач в роботі з камерою квадрокоптера використовуючи алгоритми обробки зображень.

Мета кваліфікаційної роботи – розроблення трекінгу об'єктів з використанням комп'ютерного зору квадрокоптера шляхом розробки програмного забезпечення на базі C++ OpenCV.

Об'єкт дослідження – процес трекінгу об'єктів на основі комп'ютерного зору квадрокоптера.

Предмет дослідження – методи обробки зображень, ітераційний алгоритм CAMShift.

Методи дослідження – аналіз алгоритму обробки вхідного зображення CAMShift через комп'ютерний зір квадрокоптера для реалізації трекінгу об'єктів.

Результатом кваліфікаційної роботи є розроблення програмної частини трекінгу об'єктів використовуючи методи обробки зображення бібліотеки OpenCV та алгоритму CAMShift. Розроблена програма має можливість відслідковувати над вибраною областю об'єкта у різних умовах освітленості.

Проведений огляд різних типів квадрокоптерів, технічних засобів які використовуються для реалізації комп'ютерного зору, їх використання в різних умовах. Аналіз методів обробки зображень, огляд інтелектуальних систем для комп'ютерного зору використання методів алгоритмів трекінгу в суміжності з комп'ютерним зором.

Для вирішення поставленого завдання необхідно:

- провести аналіз типів квадрокоптерів, їх технічні засоби та їх використання;
- провести аналіз методів обробки зображень, алгоритмів та інтелектуальних систем для використання їх у комп'ютерному зорі квадрокоптера;
- розробити програмне рішення для виконання трекінгу об'єкта використовуючи комп'ютерний зір квадрокоптера;
- провести експериментальну частину розробки та зробити виновки роботи програми;
- оформити кваліфікаційну роботу згідно ДСТУ 3008:15 [1], а також з методичними вказівками з підготовки і оформлення кваліфікаційної роботи [2].

Отримані результати можна віднести до Цілі сталого розвитку 11 “Сталий розвиток міст і громад”, а саме п.11.4 “Забезпечити своєчасне оповіщення населення про надзвичайні ситуації з використанням інноваційних технологій”.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВА ЗАДАЧІ

1.1 Постановка задачі кваліфікаційної роботи

Квадрокоптер – це апарат, важчий за повітря, здатний до вертикального зльоту і посадки, який рухається в повітрі чотирма гвинтами, розташованими в одній площині. На відміну від звичайних вертольотів, квадрокоптер може змінювати швидкість обертання кожного гвинта, чим і забезпечується його політ. Два з них обертаються за годинниковою стрілкою, два – проти, що дозволяє не тільки рухати апарат у горизонтальній чи вертикальній площині, а й нахилити його в різні боки [3].

Типові квадрокоптери – це невеликі безпілотні апарати. Основною частиною квадрокоптера є рама з чотирма подовженнями, на кінцях яких кріпляться гвинти. Каркас апарата легкий і при цьому жорсткий, у ньому розміщуються акумулятор для живлення електродвигунів постійного струму з гвинтами, плата контролера та сигнальні вогні. Акумулятор зазвичай розташовується в нижній частині апарата задля його стабільності під час польоту. Двигуни розташовані на однаковій відстані від центру [3].

Хоча квадрокоптери почали розроблятися в 1920-і, а особливо в 1950-ті роки, вони не могли конкурувати зі звичайними вертольотами чи конвертопланами. Поширення квадрокоптерів відбулося тільки з 2000-х років, коли розвиток мікроелектроніки дозволив реалізувати квадрокоптери в формі БПЛА або дронів [3].

Людина, що керує безпілотним квадрокоптером за допомогою віддаленого пульта, називається оператором. Зв'язок з оператором відбувається через Bluetooth і/або Wi-Fi. Квадрокоптер може мати автоматизовані системи зльоту/посадки, що не потребують контролю з боку людини. Деякі квадрокоптери також здатні літати цілком автономно [3].

1.1.1 Класифікація квадрокоптерів

З розвитком мікроконтролерів в розробці БПЛА в повсякденному житті з'явилися безліч варіантів безпілотників які мають виконувати різні види роботи та мають свої аспекти в конструкції та управлінні.

Квадрокоптери в більшості класифікуються за різними критеріями, кількістю та розташування гвинтів або крил, мети використання, дальності польоту, масі корисного навантаження, базового механізму.

За типом конструкції:

- трикоптери оснащений 3 гвинтами;
- квадрокоптери із 4 гвинтами;
- гексакоптери із 6 гвинтами;
- октокоптери із 8 гвинтами.

Класифікація за технічними характеристиками:

- вертольотного/коптерного (мультироторні – мультикоптерні та співвісні);
- літакового (з нерухомим крилом);
- гібридного (з гвинтами та крилами).

Класифікація за розміром:

- БПЛА важать менше 10 кг, максимальний час перебування у повітрі 60 хвилин;
- вага БПЛА сягає 50 кг, час перебування у повітрі сягає 5 годин;
- безпілотні літальні апарати вагою до 1 тонни на 15 годин польоту;
- важкі безпілотники.

Повний рисунок більшості типів БПЛА можна побачити на рис. 1.1.

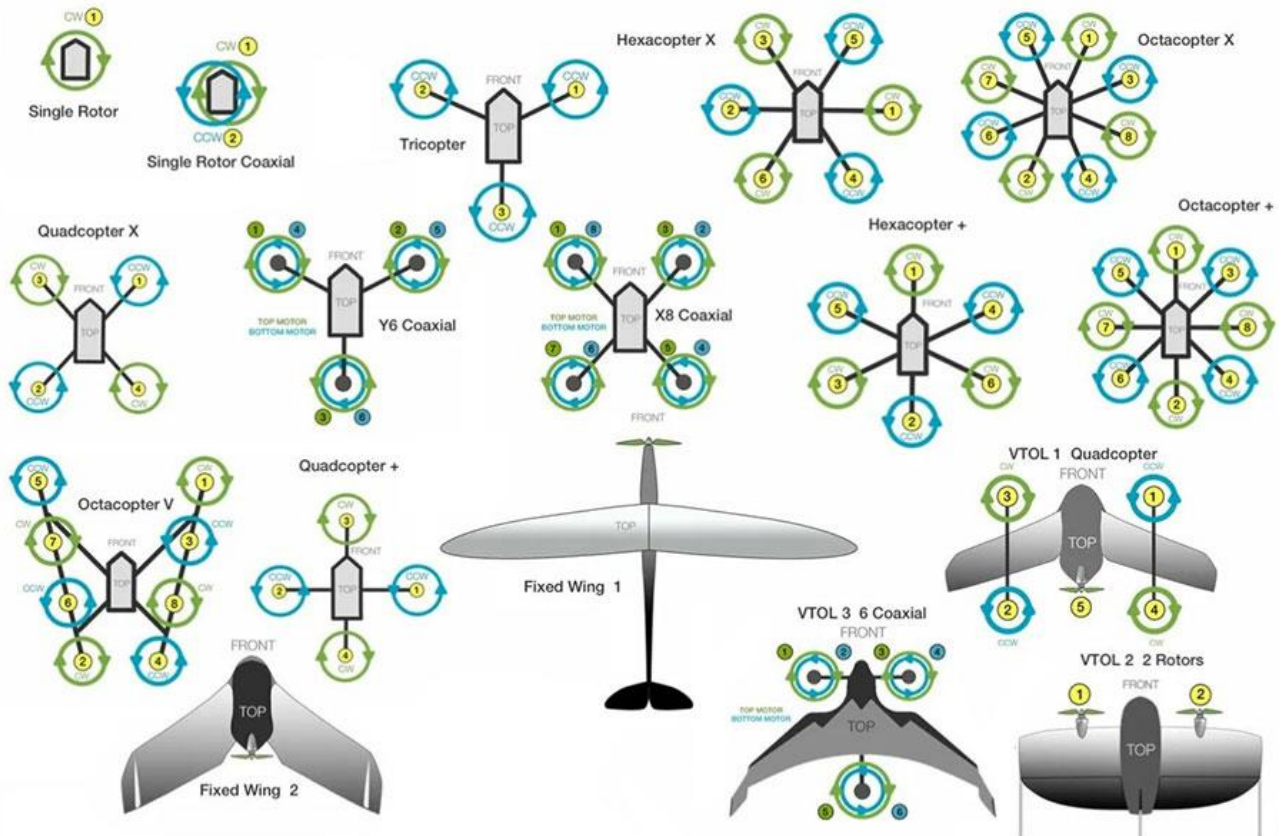


Рисунок 1.1 – Класифікація дронів за технічними характеристиками

Безпілотники можна також класифікувати за типом управління, в більшості вони поділяються на GPS і FPV дрони.

GPS дрони найчастіше використовуються для реалізації стабілізованої аерозйомки. Такі моделі доходять до споживача в повній, готовій до польоту з коробки комплектації (Дрон + Пульт + АКБ + ЗП) для реалізації якого достатньо прочитати інструкцію та зарядити акумулятори за допомогою зарядного пристрою, що йде в комплекті. Приклад GPS дрону можна побачити на рис. 1.2. Завдяки автоматичним функціям експлуатація GPS дронів максимально проста і не вимагає від користувача особливих навичок пілотування. Потенціал GPS дронів завжди обмежений електронікою, як у потужності, так і в управлінні, що в цілому визначає їх вузьке призначення (аерозйомка/картографія/моніторинг/пошук та порятунок).



Рисунок 1.2 – Комплект типового GPS дрону

FPV дрони (рис. 1.3), що стрімко набирають популярність, повна протилежність GPS дронам. Саме ця ніша зародила видовищні дрон-рейсинг та дрон-фрістайлінг. Такі БЛА використовуються для реалізації вільного динамічного польоту в режимі від першої особи, що дозволяє дарувати користувачеві неймовірні відчуття - польоту птиці. Згадана свобода польоту стає доступною завдяки відсутності у FPV дронів будь-яких електронних обмежень, як у потужності, так і в управлінні. В ідеалі такі дрони збираються та налаштовуються користувачем з нуля з урахуванням особистих переваг.

FPV (First Person View) квадрокоптери відрізняються від звичайних квадрокоптерів за п'ятьма основними ознаками:

- квадрокоптери оснащені камерою, яка транслює живе відео на землю, тому пілот може бачити те, що бачить квадрокоптер у першій особі;
- квадрокоптери управляються за допомогою спеціальної очкової підставки або дисплея, а не за допомогою пульта керування;

- квадрокоптери зазвичай мають вищу швидкість і прискорення, ніж звичайні квадрокоптери, що дозволяє пілоту швидше пересуватися та виконувати більш екстремальні маневри;
- квадрокоптери оснащені сучасною технологією, такою як відео передача в реальному часі та вбудовані камери, які дозволяють вам бачити все з точки зору вашого квадрокоптера.



Рисунок 1.3 – Типовий вид FPV-дрона

1.1.2 Сфера використання квадрокоптерів

З поширенням квадрокоптерів у повсякденному житті вони отримали безліч сфер застосування за рахунок своєї простої конструкції яку можна легко змінити або переробити під інший напрям роботи, якщо привести як приклад сфер використання то можна виділити такі як аерозйомка, забезпечення безпеки, екстрені служби, логістичні.

Квадрокоптер із вбудованою камерою (див. рис. 1.4) яка дозволяє переглядати відзнятий матеріал на смартфоні, ноутбучі, комп'ютері в режимі реального часу. Це дуже зручно, тому що користувач може коригувати процес

зйомки під час польоту. Дрон із камерою застосовується під час масових заходів, концертів, парадів, спортивних змагань, святкування приватних урочистостей (весілля, дня народження), корпоративу. Корисна річ для журналістів, відеооператорів, блогерів. Розширює спектр розважальних послуг – зйомка трюків на воді, повітрі тощо.



Рисунок 1.4 – Приклад дрону з професійною камерою

Квадрокоптер яке має спеціалізоване обладнання може виконувати відеоспостереження за комерційними або приватними об'єктами. Крім камер, датчиків руху, в охоронних цілях застосовуються дрони. Апарати здатні патрулювати периметр, передавати картинку оператору. Техніка може бути доукомплектована тепловізорами чи приладами нічного бачення, що значно покращує функціонал.

Приклад дрону спостереження можна побачити на рис. 1.5.



Рисунок 1.5 – Автономний дрон для охорони та патрулювання

Багато людей за допомогою коптерів вигулюють своїх домашніх вихованців. Актуально для приватного будинку. У сільському господарстві безпілотники використовуються для спостереження за худобою. Сучасні апарати роблять знімки полів, що допомагає моніторити дозрівання врожаю. Вносять добрива, обприскують, створюють 3D-картки. Це загалом економить час та кошти аграрія. Безпілотники здатні працювати за непередбачених погодних умов. Приклад аграрного дрону представлений на рис. 1.6.



Рисунок 1.6 – Аграрний дрон в дії

1.2 Огляд технічних засобів комп'ютерного зору

Камера в квадрокоптері є одним з ключових компонентів, залежно від якого віддається перевага тій чи іншій моделі. У її функції входить не тільки фото та відеозйомка, а й маневреність агрегату в польоті. Існує кілька різновидів камер для дронів.

FPV-камера є одним із найпоширеніших видів камер, якими оснащені безпілотники. Технологія First Person View включає спеціальні датчики (хедтрекери), які дозволяють пілоту відчувати реалістичність польоту на борту літального апарату.

FPV-камера може передавати сигнал двома способами: цифровим (за допомогою Wi-Fi) та аналоговим способом (GHz) [13].

Прикладом аналогової FPV камери можна привести FPV KaraFPV SPY 1200TVL (SPY) (див. рис. 1.7).



Рисунок 1.7 – Аналогова камера FPV KaraFPV SPY 1200TVL (SPY)

Переваги:

- має миттєвий відгук кадрів;
- робота при слабкому освітленні.

Недоліки:

- низька якість зображення;
- обмежені налаштування.

Прикладом цифрової FPV камери був взятий Foxeer T Rex Micro 1500TVL (HS1252) (див. рис. 1.8)



Рисунок 1.8 – Цифрова камера FPV Foxeer T Rex Micro 1500TVL (HS1252)

Переваги:

- краща чіткість зображення;
- має кращу спроможність в модифікації.

Недоліки:

- вища ціна;
- існує застримка сигналу.

Відеокамера для квадрокоптера - це, як правило, вбудована в квадрокоптер камера, призначена для зйомки високоякісних зображень та відео. Такі камери мають низку переваг для організації відеозйомки на професійному рівні [13]:

- роздільна здатність досягає якості UltraFHD 4K, що забезпечує високу деталізацію зображення;
- мають найкращу стабілізацію зображення;
- мають різні стандарти стиснення, наприклад, H.264 та H.265, що впливає на якість відео та бітрейт.

Прикладом стандартного квадрокоптера з вбудованою камерою можна привести DJI Mavic Pro (див. рис. 1.9).



Рисунок 1.9 – Квадрокоптер з вбудованою камерою DJI Mavic Pro

Вбудовані камери більш легкі в використанні через інтеграцію з системами самого квадрокоптера та мають вбудовану стабілізацію зображення, що дозволяє виводити всю інформацію про політ на зображення з камери та легко керувати самим квадрокоптером.

Екшн-камера для дрона – це невеликий і міцний пристрій, що має чудову водонепроникну властивість, а також здатний витримувати високі швидкості та вібрації в польоті.

Екшн-камери відрізняються за деякими критеріями:

- стійкі до негоди, сильних вібрацій, ударостійкі;
- мають широкий кут зору: від 170° до 360°;
- забезпечують кількість кадрів до 120 за секунду.

Тепловізійна камера для квадрокоптера є додатковим обладнанням, яке зчитує температури досліджуваних поверхонь. Такі камери оснащені об'єктивом особливої конструкції та датчиками, які здатні вловлювати інфрачервоне випромінювання та пропускати його через себе.

1.3 Висновки аналізу предметної області

На першому розділі було розглянуто поняття квадрокоптера, як їх класифікувати та як їх можна використовувати, розглянуті технічні засоби для реалізації комп'ютерного зору квадрокоптера.

За класифікацією ми в результаті маємо безліч типів дронів які мають унікальну конструкцію, з методами використання виявляється, що майже в кожному типі квадрокоптера є відеокамера яка виконує різні функції від навігації, огляду місцевості або зйомки відео.

Проаналізувавши всі конструкції та методи використання можна зробити висновок, що в усіх цих дронів є реалізація комп'ютерного зору з різним рівнем автоматизації.

Отже, на основі цих систем комп'ютерного зору можна розробити свій алгоритм трекінгу об'єктів для покращення виявлення або подальшій навігації.

2 АНАЛІЗ СУЧАСНИХ МЕТОДІВ КОМП'ЮТЕРНОГО ЗОРУ

2.1 Системи комп'ютерного зору

Із сучасним швидким рухом технологічного прогресу, різні системи комп'ютерного зору стали невід'ємною частиною нашого життя. Наші смартфони, системи сигналізації, автомобілі та багато інших об'єктів, що нас оточують, вже не можуть функціонувати повноцінно без них.

Комп'ютерним зором називають галузь знань, яка об'єднує методики обробки, аналізу та розпізнавання зображень, а також великих масивів даних з фізичного виміру з метою отримання певних результатів (інтерпретованих чисельних або символічних) [7].

Реалізація систем комп'ютерного зору сильно залежить від галузі їх застосування, апаратної платформи і вимог по продуктивності. Деякі системи є автономними і вирішують специфічні проблеми детектування і вимірювання, тоді як інші системи складають підсистеми більш великих систем, які вже можуть містити підсистеми контролю механічних маніпуляторів (роботи), інформаційні бази даних (пошук схожих зображень), інтерфейси людинамашина і (комп'ютерні гри) т.д. Однак, існують функції, типові для багатьох систем комп'ютерного зору [8].

Класична задача в комп'ютерному зорі, обробці зображень і машинному зорі це визначення чи містять віддання деякий характерний об'єкт, особливість або активність. Це завдання може бути достовірно і легко вирішена людиною, але до цих пір не вирішена задовільно в комп'ютерному зорі в загальному випадку: випадкові об'єкти у випадкових ситуаціях.

Існуючі методи вирішення цього завдання ефективні тільки для окремих об'єктів, таких як прості геометричні об'єкти (наприклад, багатогранники), людські обличчя, друковані чи рукописні символи, автомобілі і тільки в

певних умовах, звичайно це певне освітлення, фон і положення об'єкта відносно камери.

Прикладами застосування систем комп'ютерного зору можуть бути:

- системи управління процесами (промислові роботи, автономні транспортні засоби);
- системи відео нагляду;
- системи організації інформації (наприклад, для індексації баз даних зображень);
- системи моделювання об'єктів або навколишнього середовища (аналіз медичних зображень, топографічне моделювання);
- системи взаємодії (наприклад, пристрої введення для системи людиномашинного взаємодії);
- системи доповненої реальності;
- обчислювальна фотографія, наприклад для мобільних пристроїв з камерами.

Приклад застосування систем комп'ютерного зору для класифікації об'єктів наведено на рис. 2.1



Рисунок 2.1 – Візуалізація роботи системи класифікації об'єктів з використанням комп'ютерного зору камери

2.2 Аналіз методів обробки зображень

З використанням відеокамери яка прикріплена до квадрокоптера в польових умовах або дрону в більшості випадків буде мати погіршене зображення через вплив погодних умов, перешкод зв'язку або через перенасичення або нестачі світла. Тому для вирішення цієї проблеми використовують вбудовані або зовнішні методи обробки зображень для покращення зору користувача або програми яка використовує камеру як систему комп'ютерного зору для навігації або трекінгу.

Метод ідентифікації об'єктів, що знаходяться в робочій зоні дрону, зазвичай включає два етапи: виділення характерних ознак об'єктів; власне розпізнавання об'єктів по знайденій сукупності характерних ознак. Відповідно з такою структурою процесу ідентифікації алгоритми обробки інформації в системі технічного зору прийнято ділити на алгоритми попередньої обробки і алгоритми розпізнавання, що носить певною мірою умовний характер, так як в деяких практичних додатках одні й ті ж з математичної суті алгоритми можуть бути використані на обох етапах розглянутого процесу.

Перед тим, як методи комп'ютерного зору можуть бути застосовані до відеоданих з тим, щоб витягти певну частку інформації, необхідно обробити відеодані, з тим щоб вони задовольняли деяким умовам, залежно від використовуваного методу.

Наприклад для покращення зображення який в результаті передачі має перешкоди у вигляді шуму можна вирішити завдяки фільтрам шуму (рис. 2.2)

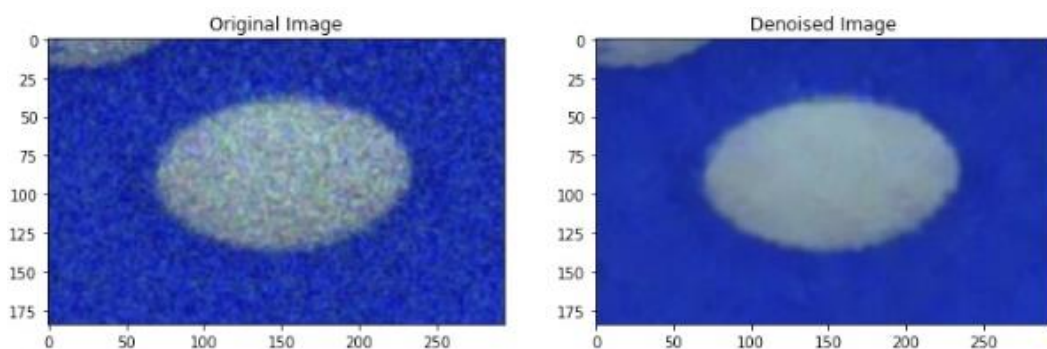


Рисунок 2.2 – Приклад використання фільтра шуму

Приклади попередньої обробки є:

- повторна вибірка з тим, щоб переконатися, що координатна система зображення вірна;
- видалення шуму з тим, щоб видалити спотворення, що вносяться датчиком;
- поліпшення контрастності, для того, щоб потрібна інформація могла бути виявлена;
- масштабування для кращого розрізнення структур на зображенні.

Для кращого використання зображення у комп'ютерному зорі в більшості випадків використовують інші колірні моделі зображення, наприклад HSV (рис. 2.3).

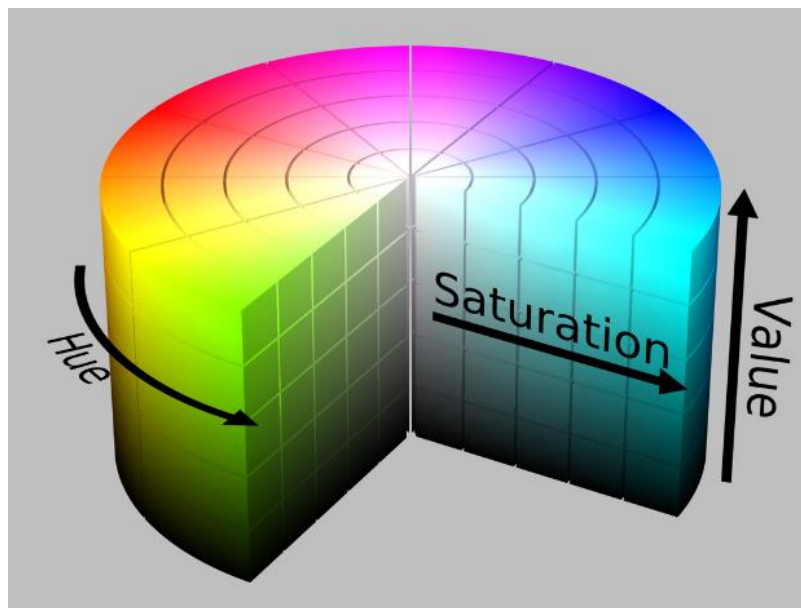


Рисунок 2.3 – Колірна модель HSV

HSV (Hue, Saturation, Value) являє собою колірну модель яка описує їх завдяки відтінку, насиченості та яскравості. Відміну від моделі RGB, вона має переваги в обробці та детекції зображень тому, що вона більш близька до людського бачення кольорів. Завдяки цьому система комп'ютерного зору має більше інформації для виділення ознак та подальшого трекінгу.

Приклад порівняння зображення з різними кольоровими моделями представлений на рис. 2.4.

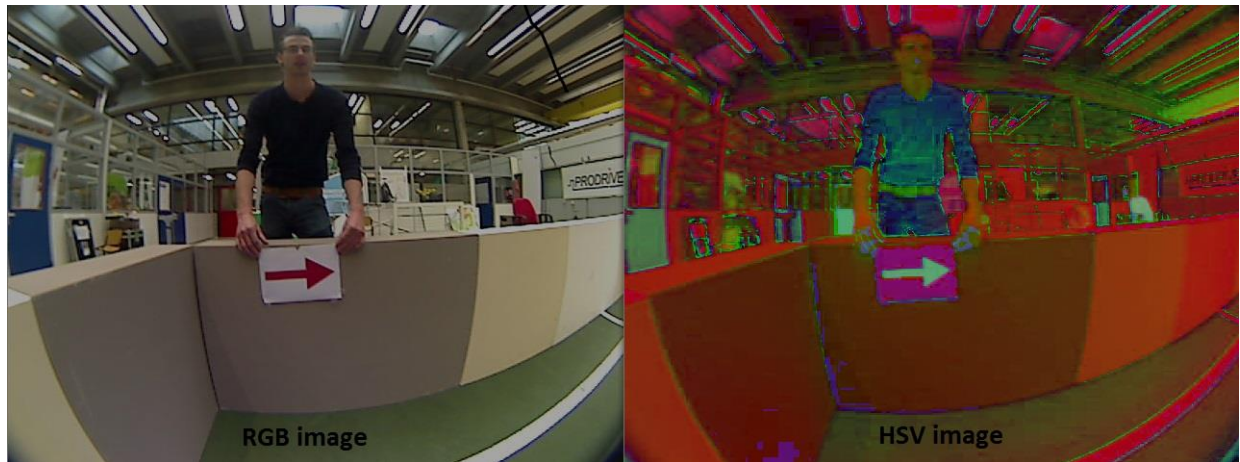


Рисунок 2.4 – Порівняння RGB та HSV зображення

Для поліпшення використання та передачі даних у роботі з комп'ютерним зором можна використовувати методи виділення меж та ліній.

Метод Кенні використовується для виділення ліній як корисної інформації з зображення для зниження значного обсягу даних для обробки.

Загальними принципами методу Кенні для виділення ознак є [8]:

- виявлення меж з низьким рівнем похибок;
- визначення центральних точок визначених меж;
- позначення меж у вихідному зображенні із мінімізацією шумів.

У загальних рисах алгоритм Кенні складається з 5 кроків:

- використання фільтра Гауса для згладжування зображення та зменшення шумів;
- знаходження градієнтів інтенсивності на зображенні;
- застосування немаксимального пригнічення вихідних значень меж;
- застосування подвійного порогового метода для визначення

потенційних меж;

- визначення меж за гістерезисом, закінчення виявлення меж пригніченням слабких та незв'язаних елементів меж об'єктів.

Приклад методу Кенні представлений на рис. 2.5.

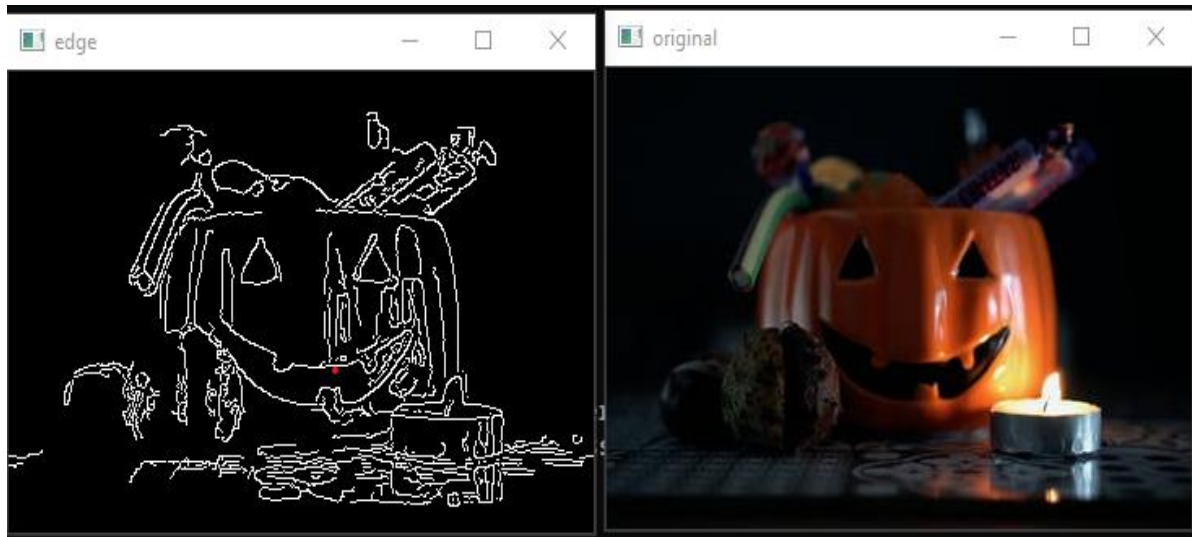


Рисунок 2.5 – Реалізація методу Кенні для визначення меж зображення

Виділення деталей: деталі зображення різного рівня складності виділяються з відеоданих. Типовими прикладами таких деталей є:

- лінії, межі і кромки;
- локалізовані точки інтересу, такі як кути, краплі або точки: більш складні деталі можуть ставитися до структурою, формою або руху.

На певному етапі обробки приймається рішення про те, які точки або ділянки зображення є важливими для подальшої обробки, тобто проводиться сегментація.

Прикладами сегментації є:

- виділення певного набору цікавлять точок;
- сегментація одного або декількох ділянок зображення, які містять характерний об'єкт.

На етапі високорівневій обробці вхідні дані зазвичай представляють невеликий набір даних, наприклад набір точок або ділянку зображення, в якому імовірно знаходиться певний об'єкт.

Прикладами висорівневої обробки є:

- перевірка того, що дані задовольняють умовам, що залежать від методу і застосування;
- оцінка характерних параметрів, таких як положення або розмір об'єкта;
- класифікація виявленого об'єкту по різних категоріях.

Тобто для більшості задач для ідентифікації, обробки зображень, покращення або виділення ознак використовуються прості форми комп'ютерного зору які в основі базуються на алгоритмах цифрової обробки зображень, які виконують покроковий аналіз пікселів, контурів, текстур і кольорів зображення. Такі методи не потребують великих обчислювальних ресурсів.

Основні етапи роботи комп'ютерного зору з використанням алгоритмів можна назвати:

- попередня обробка зображення(фільтрація шумів, нормалізація, підвищення контрасту);
- конвертація колірної моделі зображення(BGR, HSV, GRAYSCALE);
- виділення контурів та ліній (методи Кенні, оператор Собеля, Хафа);
- виділення ознак(використання алгоритмів які визначають характерні точки об'єктів).

Для реалізації алгоритму трекінгу об'єкту з використанням комп'ютерного зору можна використати алгоритм CAMShift (Continuously Adaptive Mean Shift).

CAMShift являє собою алгоритм кластеризації, який сегментує зображення та відстежує кластери даних шляхом пошуку областей з найвищою щільністю точок. Основний принцип його роботи полягає в постійному русі кожної точки у напрямку максимальної щільності, що дозволяє алгоритму визначати кількість точок та форму кластерів.

Приклад роботи алгоритму CAMShift представлений на рис. 2.6.

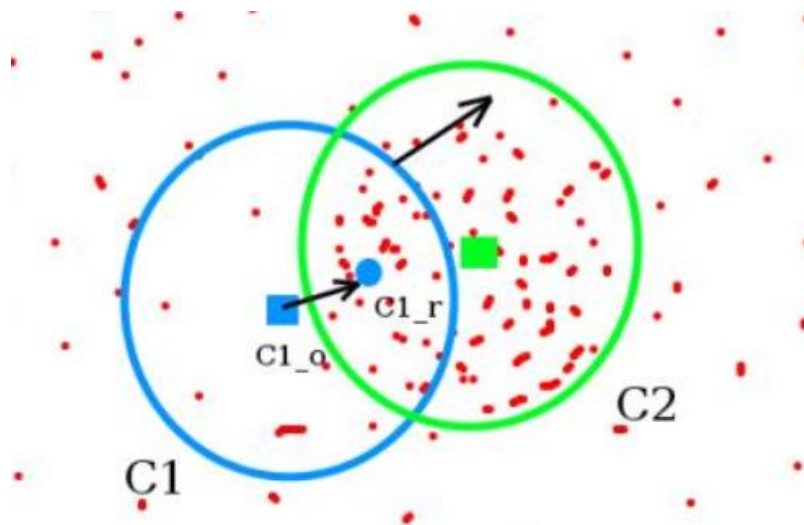


Рисунок 2.6 – Ілюстрація роботи алгоритму CAMShift

На рис. 2.6 проілюстровано як коло C1 та його початковий центр C1_o рухаються до першої точки C1_r який є центром мас частини кластеру точок на першій ітерації, з кожною ітерацією коло C1 буде наближатись до максимальної кількості точок який утворюють найбільший кластер який знаходиться в межах кола C2 [13].

Таким чином, якщо взяти область об'єкту та на її основі побудувати гістограму кольору, вона стає основою для відстеження змін освітлення. Наприклад, якщо взяти H за колір гістограми в обраній області, то гістограма визначається за (2.1) як розподіл частоти для різних кольорів з точки зору відтінку, насиченості та яскравості (HSV):

$$H(h) = \frac{\text{count}(h)}{\text{total}_{\text{count}}} \quad (2.1)$$

де H – колір гістограми в обраній області;

h – значення кольору;

$\text{count}(h)$ – число пікселів з кольором h ;

$\text{total}_{\text{count}}$ – число всіх вибраних пікселів.

Обчислення зображення зворотної проєкції P (2.2) обчислюється для кожного нового кадру I .

$$P(x, y) = H(I(x, y)) \quad (2.2)$$

де P – зворотна проєкція;
 (x, y) – координата пікселю;
 $I(x, y)$ – значення кольору пікселю на зображенні.

Алгоритм середнього зсуву в CAMShift шукає максимум розподілу ймовірностей на зображенні зворотної проєкції. Він визначає новий центр області пошуку за допомогою наступного виразу:

$$m = \frac{\sum_{x,y} p(x, y)c(x, y)}{\sum_{x,y} p(x, y)} \quad (2.3)$$

де m – новий центр області пошуку після виконання MeanShift;
 $p(x, y)$ – значення зворотної проєкції для пікселю (x, y) ;
 $c(x, y)$ – координати пікселю.

Таким чином алгоритм крок за кроком “слідкує” за виділеним об’єктом, приклад трекінгу можна побачити на рис. 2.7.



Рисунок 2.7 – Візуалізація роботи алгоритму без адаптації форми

Для візуалізації адаптації форми вікна потрібно обчислити розмір (2.4):

$$s = 2 * \sqrt{\frac{M_{00}}{256}} \quad (2.4)$$

де s – розмір вікна;

M_{00} – момент нульового порядку;

Завдяки цьому тепер можна відобразити зміну розміру вікна за об'єктом за яким ведеться трекінг (рис. 2.8.).



Рисунок 2.8 – Візуалізація зміни розміру вікна трекінгу відносно об'єкту

2.3 Огляд інтелектуальних систем комп'ютерного зору

Інтелектуальний комп'ютерний зір представляє собою використання систем на глибокому навчанні і нейронних мереж, які спроможні самостійно навчатися розпізнавати закономірності в зображеннях.

Під інтелектуальною системою розуміють об'єднану інформаційним процесом структуру технічних засобів і програмного забезпечення, котрі працюють у взаємозв'язку з оператором або незалежно від нього; здатну на основі відомостей і знань при наявності мотивації синтезувати, генерувати рішення про дію та знаходити ефективні шляхи вирішення завдань.

Використання штучних нейронних мереж дозволяє розв'язати задачі керування нелінійними ОК шляхом створення адаптивних САК з навчаємим нейрорегулятором.

Як приклад для можна взяти згорткову нейронну мережу, вона є алгоритмом, який отримує вхідне зображення та призначає вагові значення візуальним об'єктам в ньому, щоб визначити їх належність до певних класів та мати можливість відрізнити один об'єкт від іншого. Завдяки згортковому алгоритму нейроні мережі здатні фіксувати просторові та часові залежності на зображенні та навчитися визначати зміст зображення навіть краще людини.

2.4 Застосування методів систем комп'ютерного зору

В зовнішніх умовах в залежності від камери квадрокоптера можна часто зустріти шуми, розмиття та засвітлення зображення. Тому використовуються методи обробки зображень, корекції яскравості й контрастності. В більшості моделях квадрокоптерів можна зустріти попередню обробку зображення.

Використання інтелектуальних методів систем комп'ютерного зору квадрокоптера значно поліпшує загальний контроль над БПЛА та дає змогу автоматизувати дії в залежності від потреб.

Інтелектуальні системи можуть застосовуватися для розпізнавання та навігації в навколишньому середовищі за допомогою технологій комп'ютерного зору, LIDAR, GPS та інерціальних вимірювальних одиниць (IMU). Такий підхід забезпечує здатність безпілотного літального апарата аналізувати рельєф місцевості, будувати оптимальні траєкторії руху, а також визначати й відстежувати задані об'єкти або області спостереження [11].

Тобто, завдяки інтелектуальним системам на основі нейромереж можна досягти більшої автономності в управлінні та контролі квадрокоптером або БПЛА в складних умовах.

2.5 Висновки аналізу методів комп'ютерного зору

В другому розділі ми проаналізували різні методи обробки зображень та зробили огляд інтелектуальних систем. Для подальшої розробки системи трекінгу об'єктів був обраний алгоритм CAMShift для подальшого дослідження в різних умовах.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТРЕКІНГУ ОБ'ЄКТІВ

3.1 Вибір програмної середи

Для програмної реалізації алгоритму CAMShift за основу взята мова програмування C++ та програмна середа Visual Studio 2026.

Visual Studio широко використовується для розробки програмного забезпечення на мовах C++, C# та інших. Воно поєднує в собі редактор коду, компілятор, засіб керування проектами та інструменти аналізу, що дозволяє розробнику повністю контролювати розробку в межах одного робочого простору.

Програмна середа Visual Studio 2026 має в собі безліч зручних бібліотек для розробок як підтримка C++/CLI, .NET Framework, MSVC, MFC-бібліотек. Також вона має можливість в підключенні сторонніх бібліотек як приклад популярна бібліотека OpenCV яка буде використовуватись для реалізації трекінгу об'єктів.

3.2 Програмна реалізація трекінгу об'єкту

Програмна реалізація була зроблена з використання як основи платформи Windows Forms для більш зручного виводу зображень камери та налаштувань параметрів програми.

Windows Forms являє собою платформу бібліотеки .NET яка має можливості в розробці класичних програм із графічним інтерфейсом Windows, вона дозволяє швидко й ефективно створювати графічний інтерфейс та елементи управління які можуть бути використанні для управління та тестування програми.

Був створений графічний інтерфейс який виконується завдяки основному файлу програми myForm.cpp, він посилається на заголовочний файл myForm.h який містить в собі реалізацію графічного інтерфейсу та трекінгу об'єкта.

Програмний код myForm.cpp представлений на рисунку 3.1.

```
1      #include "MyForm.h"
2
3      using namespace System;
4      using namespace System::Windows::Forms;
5
6      [STAThreadAttribute]
7
8      void Main()
9      {
10         CAMShift::MyForm form;
11         Application::Run(% form);
12     }
```

Рисунок 3.1 – Програмна реалізація myForm.cpp

В програмі ми виконуємо запуск функції Main в одному потоці яка має посилання із іншого файлу MyForm.h.

В файлі MyForm.h програма починається з посилань на окремі файли бібліотеки OpenCV 2.1 (див. рис. 3.2).

```
#pragma once
#pragma comment(lib,"cv210.lib")
#pragma comment(lib,"cvaux210.lib")
#pragma comment(lib,"cxcore210.lib")
#pragma comment(lib,"highgui210.lib")

#include <iostream>
#include <highgui.h>
#include <cv.h>
```

Рисунок 3.2 – Посилання на файли бібліотек OpenCV та iostream

Для роботи програми ми виконуємо оголошення всіх основних змінних які будуть використовуватися для реалізації програми (див. рис. 3.3).

```
CvCapture* capture;
IplImage* image = 0, * hsv = 0, * hue = 0, * mask = 0, * backproject = 0, * histimg = 0;
CvHistogram* hist = 0;
int backproject_mode = 0;
int select_object = 0;
int track_object = 0;
int show_hist = 1;
CvPoint origin;
CvRect selection;
CvRect track_window;
CvBox2D track_box;
CvConnectedComp track_comp;
int hdims = 16;
float hranges_arr[] = { 0,180 };
float* hranges = hranges_arr;
int vmin = 10, vmax = 256, smin = 30;
void OnCamera();
void on_mouse(int event, int x, int y, int flags, void* param);
CvScalar hsv2rgb(float hue);
```

Рисунок 3.3 – Оголошення змінних в файлі myForm.h

На рисунку 3.3 ми оголошуємо змінні для роботи захвату кадрів, створення гистограми, режимів роботи програми та інструменти для роботи вікна вибору об'єкта для трекінгу. Також задаються функції OnCamera() яка буде містити в собі основну програму трекінгу, on_mouse() буде містити в собі програму вибору об'єкта завдяки миші комп'ютера.

Для роботи програми ми завдяки функціям MyForm(), InitializeComponent() та button1_Click() створюємо графічні елементи нашої програми. На рис. 3.4 представлений результат роботи функцій.

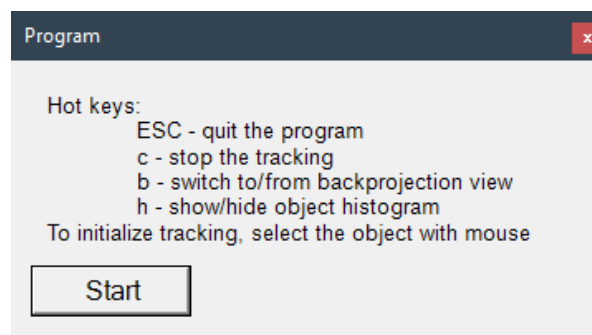


Рисунок 3.4 – Графічне вікно програми

Програмне вікно включає в собі інформацію для управління програмою та кнопку яка запускає й виводить зображення з відеокамери.

Функція `button1_Click` виконує запуск захвату кадрів з відеокамери яка під'єднана до комп'ютера, та якщо захват кадрів успішний то вона почне виводити інформацію об управлінні програмою та почне виконувати функцію `OnCamera()`.

Програмна реалізація кнопки представлена на рис 3.5.

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{
    if (capture = cvCaptureFromCAM(0))
    {
        System::String^ s;
        s = System::String::Format("Hot keys: \n"
            "\tESC - quit the program\n"
            "\tc - stop the tracking\n"
            "\tb - switch to/from backprojection view\n"
            "\th - show/hide object histogram\n"
            "To initialize tracking, select the object with mouse\n");
        Graphics^ g = MyForm::CreateGraphics();
        g->DrawString(s, gnew Drawing::Font("Arial", 10), Brushes::Black, 20, 20);

        OnCamera();
    }
    else MessageBox::Show("Camera reading Error");
};
```

Рисунок 3.5 – Код функції `button1_Click()`

Функція `OnCamera()` спершу виконує створення вікон гістограми, виводу відеокамери та треків для управління параметрами `Vmin`, `Vmax`, `Smin` (див. рис. 3.6)

```
void OnCamera()
{
    cvNamedWindow("Histogram", 1);
    cvNamedWindow("CAMShift", 1);
    cvSetMouseCallback("CAMShift", on_mouse, 0);
    cvCreateTrackbar("Vmin", "CAMShift", &vmin, 256, 0);
    cvCreateTrackbar("Vmax", "CAMShift", &vmax, 256, 0);
    cvCreateTrackbar("Smin", "CAMShift", &smin, 256, 0);
};
```

Рисунок 3.6 – Частина коду функції `OnCamera()`

Для роботи над кадрами програма користуючись бібліотекою створює список зображень які будуть використовуватися для подальшої роботи програми(див. рис. 3.7).

```

if (!image)
{
    image = cvCreateImage(cvGetSize(frame), 8, 3);
    image->origin = frame->origin;
    hsv = cvCreateImage(cvGetSize(frame), 8, 3);
    hue = cvCreateImage(cvGetSize(frame), 8, 1);
    mask = cvCreateImage(cvGetSize(frame), 8, 1);
    backproject = cvCreateImage(cvGetSize(frame), 8, 1);
    hist = cvCreateHist(1, &hdims, CV_HIST_ARRAY, &hranges, 1);
    histimg = cvCreateImage(cvSize(320, 200), 8, 3);
    cvZero(histimg);
}

```

Рисунок 3.7 – Створені змінні зображень та гістограми

Для трекінгу об'єкту ми будемо використовувати алгоритм CAMShift. На основі вибраної області та переведення її в кольорову гістограму програма повинна слідкувати за областю з найбільшою схожістю по кольору гістограми та числом пікселів в кластері.

Саме для цього в програмі створюється тип зображення з колірною моделлю HSV. Програма беручи інформацію з зображення використовує відтінок коліру (Hue) та переводить зображення в колірну гістограму яка показана на рис. 3.8. Програмна реалізація гістограми представлена на рис. 3.9.

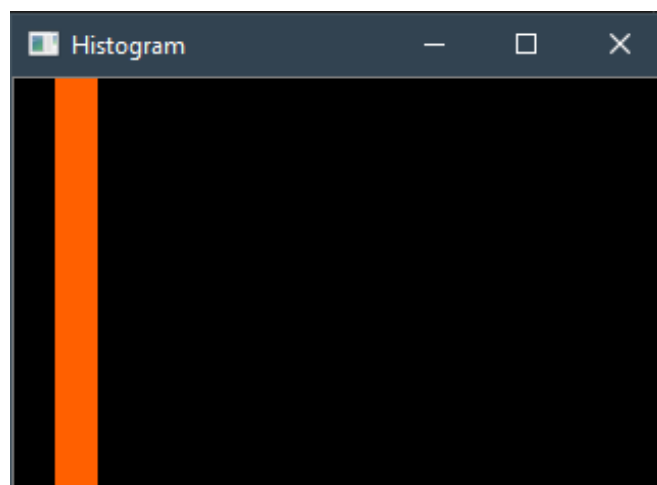


Рисунок 3.8 – Візуалізація роботи гістограми з вибраним кольором

```

if (track_object < 0)
{
    float max_val = 0.f;
    cvSetImageROI(hue, selection);
    cvSetImageROI(mask, selection);
    cvCalcHist(&hue, hist, 0, mask);
    cvGetMinMaxHistValue(hist, 0, &max_val, 0, 0);
    cvConvertScale(hist->bins, hist->bins, max_val ? 255. / max_val : 0., 0);
    cvResetImageROI(hue);
    cvResetImageROI(mask);
    track_window = selection;
    track_object = 1;

    cvZero(histimg);
    bin_w = histimg->width / hdims;
    for (i = 0; i < hdims; i++)
    {
        int val = cvRound(cvGetReal1D(hist->bins, i) * histimg->height / 255);
        CvScalar color = hsv2rgb(i * 180.f / hdims);
        cvRectangle(histimg, cvPoint(i * bin_w, histimg->height),
            cvPoint((i + 1) * bin_w, histimg->height - val),
            color, -1, 8, 0);
    }
}

```

Рисунок 3.9 – Програмна реалізація створення гистограми

Ця частина коду починає будувати та нормалізувати гистограму з використанням відтинку кольору в вибраній області. Ця гистограма потім буде використовуватися в трекінгу за допомогою розрахунку зворотної проекції.

З використанням функцій бібліотеки OpenCV ми починаємо розраховувати зворотну проекцію та на основі її буде реалізовуватися трекінг.

Частина коду відповідна за розрахунок зворотної проекції та розрахунок положення еліпсу представлена на рис. 3.10.

```

cvCalcBackProject(&hue, backproject, hist);
cvAnd(backproject, mask, backproject, 0);
cvCamShift(backproject, track_window,
    cvTermCriteria(CV_TERMCRIT_EPS | CV_TERMCRIT_ITER, 10, 1),
    &track_comp, &track_box);
track_window = track_comp.rect;

if (backproject_mode)
    cvCvtColor(backproject, image, CV_GRAY2BGR);
if (!image->origin)
    track_box.angle = -track_box.angle;
cvEllipseBox(image, track_box, CV_RGB(255, 0, 0), 3, CV_AA, 0);
}

if (select_object && selection.width > 0 && selection.height > 0)
{
    cvSetImageROI(image, selection);
    cvXorS(image, cvScalarAll(255), image, 0);
    cvResetImageROI(image);
}
cvShowImage("CAMShift", image);
cvShowImage("Histogram", histimg);

```

Рисунок 3.10 – Програмна реалізація трекінгу вибраного об'єкту

Отримуючи дані яскравості та кольорову гістограму ми виконуємо зворотну проєкцію, вона формує карту ймовірностей для пошуку максимуму пікселів до об'єкта який був вибраний. На основі карти ймовірностей починає працювати `cvCamShift()`, він автоматично починає ітеративно шукати центр маси кластеру об'єкта та оцінює його поточний розмір і орієнтацію на зображенні. Результатом роботи алгоритму стає червоний еліпс який буде слідкувати та адаптуватись під об'єкт який ми задали для трекінгу.

На рис. 3.11 представлений вид вікна де відбувається тест програми трекінгу на прикладі синьої іграшкової машини в кімнатних умовах освітлення. На рисунку 3.12 представлена кольорова гістограма об'єкта.

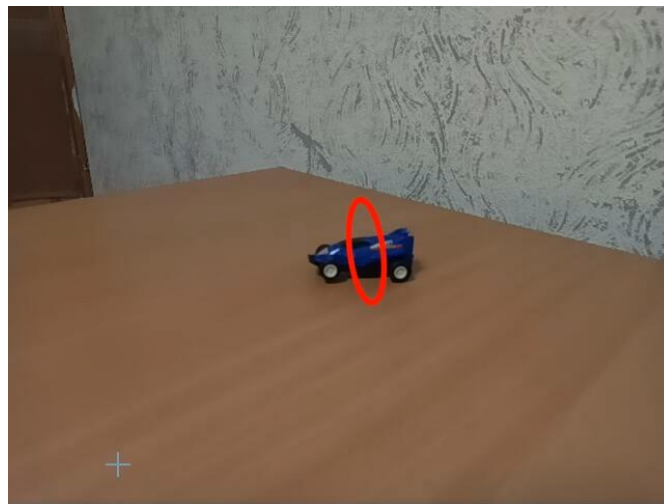


Рисунок 3.11 – Вікно CAMShift

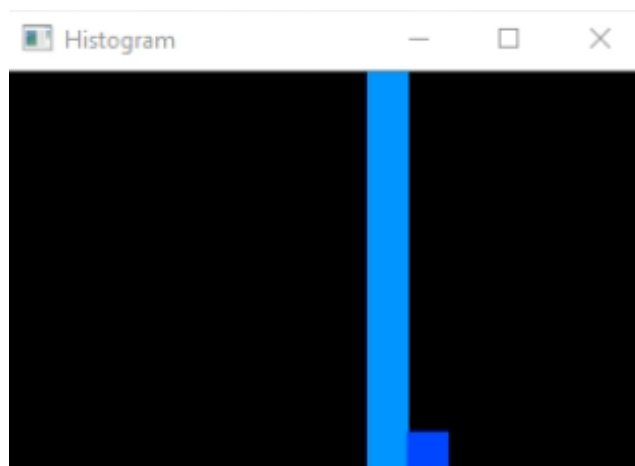


Рисунок 3.12 – Вікно кольорової гістограми

Приклад роботи програми в режимі показу зворотної проекції представлена на рис. 3.13

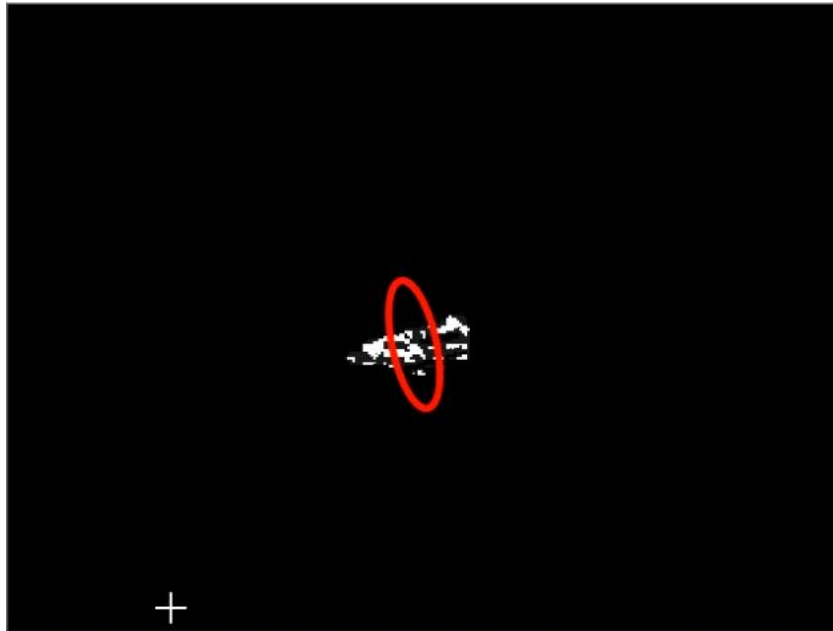


Рисунок 3.13 – Вікно SAMShift в режимі зворотної проекції

3.3 Дослідження ефективності розробленої програми

Розробивши програму з використанням ітеративного алгоритму SAMShift був проведений експеримент на якість трекінгу в різних умовах освітлення, складності об'єкта та наявності загороджень та схожих с об'єктом по формі чи кольору. В експерименті будуть використовуватись об'єкти для імітації умов, перешкоди, різні умови освітлення кімнати.

В першому експерименті був взятий об'єкт синього кольору який має більш контрастні кольори у порівнянні з фоном.

Параметри програми в продовж першого експеримента:

- $V_{min} = 10$;
- $V_{max} = 256$;
- $S_{min} = 30$.

Зображення вікна камери представлені на рис. 3.14 – 3.16.



Рисунок 3.14 – Зображення об'єкта за яким буде вестись трекінг

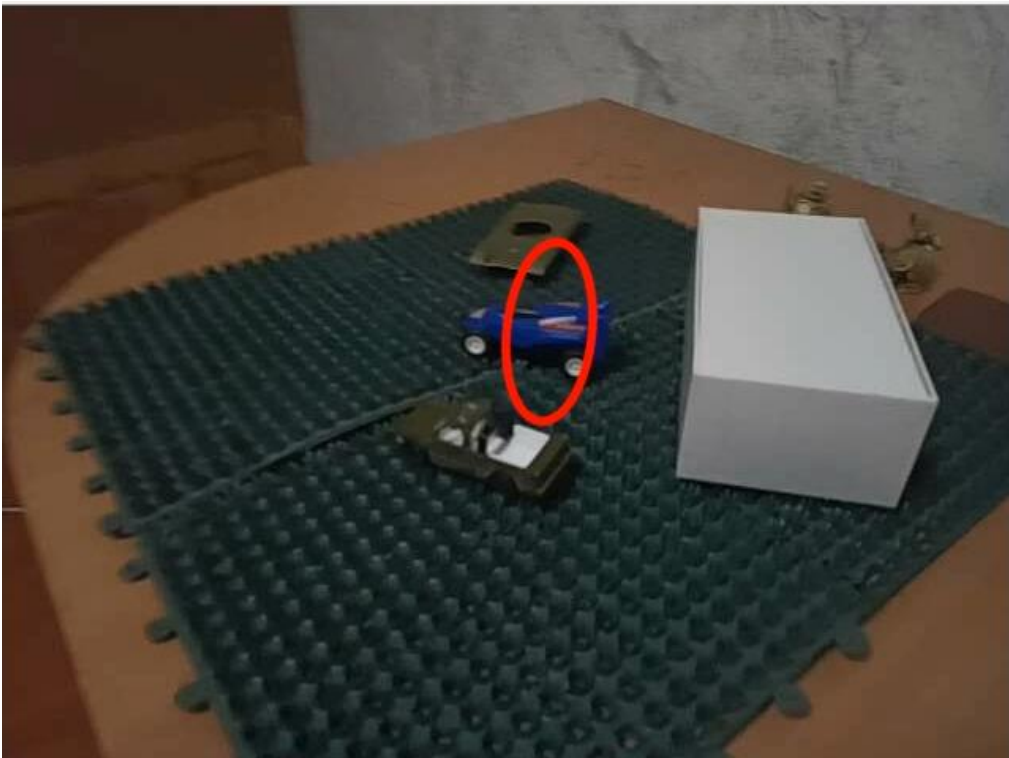


Рисунок 3.15 – Зображення трекінг об'єкта



Рисунок 3.16 – Вікно першого експерименту в режимі зворотної проекції

За результатами першого експеримента можна побачити, що програма успішно відслідковує об'єкт який був вибраний для трекінгу. Як показано на рисунку 3.16 можна побачити, що програма завдяки зображенню зворотної проекції відслідковує лише вибраний об'єкт.

За другим експериментом повинні бути виявленні межі освітлення в яких може працювати програма. Для другого експерименту був взятий об'єкт синього кольору яку програма повинна відстежувати в умовах обмеженого освітлення.

Параметри впродовж другого експеримента:

- $V_{min} = 10$;
- $V_{max} = 256$;
- $S_{min} = 30$.

Зображення вікна камери представлені на рис. 3.17 – 3.20.



Рисунок 3.17 – Трекінг об'єкта в умовах обмеженого освітлення



Рисунок 3.18 – Зворотна проекція в умовах обмеженого освітлення

Як можна бачити програма має проблеми з точним трекінгом об'єкта, через низьку освітленість на зображенні починає проявлятися шум. Для вирішення цієї проблеми було збільшені налаштування мінімальної насиченості S_{min} до 90. За цими параметрами програма почала відслідковувати об'єкт більш точноше (див. рис. 3.19, 3.20)

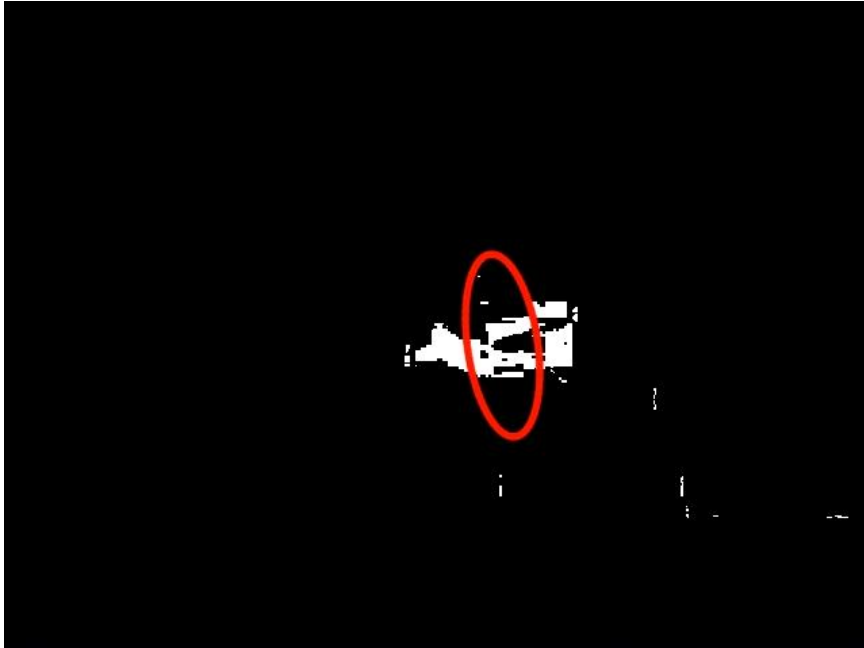


Рисунок 3.19 – Зворотна проекція зі зміненими параметрами насиченості



Рисунок 3.20 – Трекінг об'єкта зі зміненими параметрами насиченості

За результатами другого експерименту були виявлені межі за якими програма може працювати в умовах обмеженого освітлення.

Третій експеримент представляє собою тестування програми в умовах блокування об'єкта який був вибраний для спостереження. Зображення вікон третього експерименту представлені на рис. 3.21 – 3.22.



Рисунок 3.21 – Трекінг об'єкта в третьому експерименті

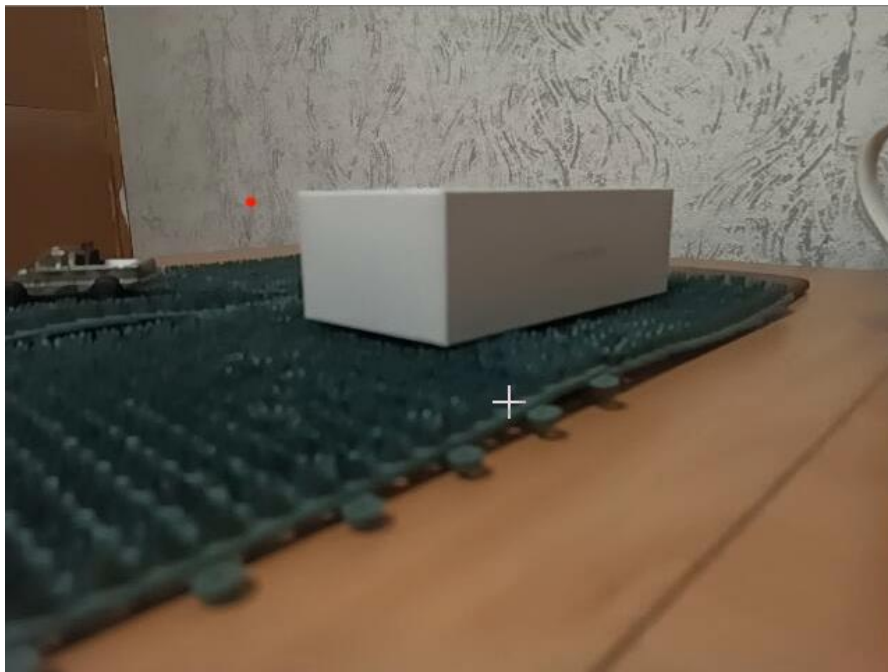


Рисунок 3.22 – Блокування трекінгу іншим об'єктом

За результатами третього експерименту можна бачити, що програма не може продовжувати відслідковувати об'єкт який знаходиться 'в тіні' іншого об'єкта. Тобто для продовження трекінгу треба мати прямий візуальний контакт з об'єктом який відстежується.

За четвертим експериментом програма буде намагатись слідкувати за зеленою машиною яка буде стояти за неповним блокуючим об'єктом у вигляді танка схожого кольору. В експерименті буде перевірено чи зможе програма відслідковувати обраний об'єкт з додаванням схожих по кольору об'єктів та неповноцінне блокування їм.

Параметри програми в продовж четвертого експеримента:

- $V_{min} = 10$;
- $V_{max} = 256$;
- $S_{min} = 30$.

Зображення вікна камери представлені на рис. 3.23 – 3.25.



Рисунок 3.23 – Вікно трекінгу четвертого експерименту



Рисунок 3.24 – Зворотна проєкція четвертого експерименту

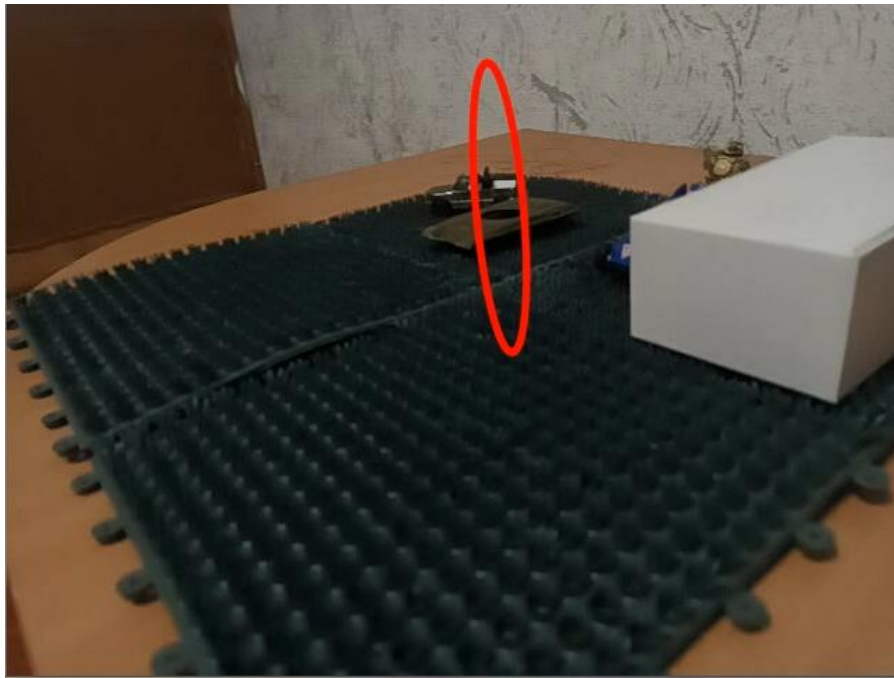


Рисунок 3.25 – Неповне блокування об'єкта



Рисунок 3.26 – Зворотна проекція з неповним блокуванням об'єкта

Як можна бачити за четвертим експериментом програма має показує значні відхилення в слідкуванні за об'єктом якщо перед ним стоїть схожий за параметрами об'єкт. У результаті якість трекінгу програми сильно падає.

3.4 Охорона праці

При використанні квадрокоптера з програмним забезпеченням який виконує трекінг об'єктів повинні бути розглянуті питання охорони праці як оператора та навколишнього середовища. Передбачено виконувати перевірку технічного стану БПЛА, акумулятора, пропелерів, камери і справності датчиків.

Основними небезпеками є:

- рухомі частини квадрокоптера;
- втрата зв'язку з дроном;
- можливість втрати управління під час зліту та посадки;
- можливість втрати управління під час польоту;
- можливість втрати управління в процесі використання програми трекінгу.

Для запобігання травм при роботі з квадрокоптером повинені бути проведений інструктаж перед використанням квадрокоптера з використанням програмного забезпечення.

Основні правила використання:

- проводити запуск, політ та посадку на безпечній від квадрокоптеру відстані;
- не доторкатися до електронних схем чи батареї під час роботи квадрокоптера;
- планувати та дотримуватись маршруту квадрокоптера згідно правилам повітряного простору України;
- проводити технічний огляд квадрокоптера до та після польоту;
- навігація та політ квадрокоптеру має пріоритет над використанням програмного забезпечення трекінгу об'єктів.

3.5 Висновок експериментальної частини

У ході роботи над програмою трекінгу об'єкту було проведено чотири експеримента які повинні дослідити можливості та ліміти програми.

За отриманими даними можна побачити, що алгоритм CAMShift має достатні показники трекінгу за об'єктом в різних умовах освітлення та може відслідковувати їх в умовах шумових перешкод. За результатами експериментів з повним блокуванням об'єкта за яким ведеться спостереження стає очевидним, що алгоритм CAMShift не має змоги відслідковувати уявний об'єкт та автоматично повертатися до нього у разі поновлення прямого контакту. Ситуація стає краще з експериментом неповного блокування об'єкта, алгоритм CAMShift має змогу поновити трекінг за об'єктом якщо поновити прямиий контакт з ним.

За цими даними можна у подальшому виправити наявні проблеми програмної реалізації з використанням інших параметрів, додаткових алгоритмів або додаванням нейронної мережі.

ВИСНОВКИ

Завданням кваліфікаційної роботи було розроблення програмної реалізації трекінгу об'єктів на основі комп'ютерного зору квадрокоптера використовуючи методи обробки зображень та алгоритму CAMShift.

Був проведений аналіз різних видів дронів та квадрокоптерів, їх технічні засоби та використання систем комп'ютерного зору. Були проаналізовані різні методи обробки зображень, алгоритми трекінгу та оглянуті інтелектуальні системи в комп'ютерному зорі. На основі аналізу була розроблена програма трекінгу яка використовує алгоритм CAMShift та камеру квадрокоптера.

На основі аналізу ітеративного алгоритму CAMShift була реалізована програма трекінгу за вибраним об'єктом. Програма була розроблена на мові C++ використовуючи бібліотеки .NET, CLI, OpenCV у програмній середі Visual Studio 2026. Програма дозволяє користувачу використовувати комп'ютерний зір квадрокоптера та використовувати його для трекінгу за вибраним об'єктом. В рамках дослідження програмна реалізація трекінгу було проведені експерименти в умовах в яких може опинитися квадрокоптер.

Були розглянуті питання охорони праці з використанням квадрокоптера з встановленим програмним забезпеченням трекінгу об'єктів. За розглянутими питаннями охорони праці були сформовані основні правила використання квадрокоптера.

На основі отриманих даних експерименту програма з використанням ітеративний алгоритму CAMShift забезпечує стабільний трекінг об'єкта в умовах обмеженого освітлення та наявності шумових перешкод. Були виявлені обмеження в роботі програмного рішення у випадках часткового та повного перекриття іншими об'єктами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. 29 с.

2. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка, освітньо-професійних програм: «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2024. 57 с.

3. Сайт Wikipedia. Сторінка «Квадрокоптер». URL:<https://uk.wikipedia.org/wiki/Квадрокоптер> (дата звернення: 28.11.2025).

4. Класифікація дронів: які види та типи бувають? *BezpekaSHOP*. URL: <https://www.bezpeka-shop.com/ua/blog/poleznyie-sovety/klassifikatsiya-dronov-kakie-vidy-i-tipy-byvayut-chastpervaya/?srsltid=AfmBOopNbsaCH5J NQm7xuk 7WoEUPpJdJH7OADK4B3273TqXSnQ1DNJ31> (дата звернення: 28.11.2025)

5. Огляд сфер використання БПЛА в повсякденному житті. *50NORTH*. URL: <https://www.50northspatial.org.ua/ua/uavs-everyday-life/> (дата звернення 28.11.2025)

6. Популярні сфери застосування квадрокоптерів. *Wondertech*. URL: <https://wondertech.ua/ua/blog/populyarni-sferi-zastosuvannya-kvadrokopteriv?srsltid=AfmBOorA6Rg7ZwaV4LDCVJce7csfXsSyD6Q-UbNYMeLdMFYIlrIDDtXE> (дата звернення: 28.11.2025)

7. Лесюк, А. М., & Яцишин, С. П. (2020). Комп'ютерний зір та його застосування. Editorial board, 530. DOI 10.46299/ISG.2020.II.VI ISBN 978-1-63649-928-4

8. Цимбал О.М. Системи адаптації роботів та технологія OpenCV: Навчальний посібник / О.М. Цимбал, А.І. Бронніков – Харків: ХНУРЕ, 2019 р. – 134 с
9. Комаркова, М. (2022). МЕТОД YOLO ДЛЯ ДЕТЕКТУВАННЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ. EDITORIAL BOARD, 403.
10. Myronuik D. M., Vlahitko B. Y., Zajazchuk I. M. Computer simulation of deep learning for image recognition. *Computer Technologies of Printing*. 2019. Vol. 2, no. 42. P. 57–63. URL: <https://doi.org/10.32403/2411-9210-2019-2-42-57-63>
11. Мамін В.А. Інтелектуальні системи керування квадрокоптерами: аналіз функціональних аспектів та перспективи розвитку. / Збірник студентських наукових статей «Автоматизація та приладобудування» «Automation and Development of Electronic Devices» ADED-2025 (Випуск 2). – 92-94 с.
12. CAMShift Algorithm for Human Tracking in the Collaborative Robot Working Area / D. Gurin, V. Yevsieiev, S. Maksymova., Ahmad Alkhalaileh // *Journal of Universal Science Research*. – 2024. – Vol. 2(8). – P. 87–101.
13. Meanshift and Camshift. *OpenCV. Open Source Computer Vision*. URL: https://docs.opencv.org/3.4/d7/d00/tutorial_meanshift.html (дата звернення: 15.12.2025)
14. Цимбал О.М. Neuro Technologies in Automation Lectures : презентації лекцій / О.М. Цимбал – Харків, ХНУРЕ.
15. Камери квадрокоптерів: які вони бувають і чим відрізняються. *UKR DRONE*. URL: <https://ukrdrone.com.ua/kameri-kvadrokopter%D1%96v-jak%D1%96-voni-buvajut-%D1%96-chim-v%D1%96dr%D1%96znajajutsja/> (дата звернення: 21.12.2025)